Design of FPGA based horizontal velocity computation and Image storage unit for Lunar Lander

Tanisha Bhatia, Jayanta Laha, Amit Maji, Arindam Mal, Subhalakshmi. K
(*Scientist*) Laboratory For Electro Optics Systems
Indian Space Research Organization
Bangalore, India
jslakshmi@leos.gov.in

Abstract- **Space field has experienced vigorous advancement with respect to evolution of vision system, image storage and processing. Real time image processing has become one of the most important tools for navigation and landing for planetary and lunar missions. Information of horizontal velocity with high accuracy will be required to do accurate pin point landing. For the testability of such a system as well as to have the understanding of Lander dynamics, prior landing image sequences are required to initially testing the algorithm [1]. This paper deals with FPGA based processing on Image sequence to find the relative velocity and also implements Image Storage in a microSD card. Landing sequence is stored in SD card and post landing they are downloaded. These images provide a very useful information about lighting, lander dynamics to fine tune algorithm for future mission. Besides this image data serves the testability during various phases of testing during development.**

Keywords—Phase correlation; RTAX2000S FPGA, FFT, SD card, Active Pixel Sensor

I. Introduction

In recent times, lunar and interplanetary exploration and pin point landing in unknown terrain is gaining importance. Continuous knowledge of spacecraft's position and velocity is needed for its proper navigation as well as soft landing [9]. Lander not only has downward vertical velocity but also relative horizontal velocity with respect to the terrain of the celestial body. The soft landing requires the correct estimate of horizontal velocity so as to kill it by thruster firing. Hence the horizontal velocity has to be accurately measured and controlled. With the advent of image processing based techniques in hardware , vision based navigation is becoming a popular option for planetary rover and lander. Lander horizontal velocity can be found using on board accelerometer but it requires initialization. Camera based velocity determination can be used to correct IMU data periodically. Vision based velocity sensor has low power and less mass than other sensors. So imaging based velocity computation is becoming an obvious choice for lunar Lander. In this method, two consecutive frames are captured with a fixed delay and pixel shifts are computed. Then pixel resolutions are converted to distance with altimeter data. Thus velocity computation requires to find the pixel shift between the frames which is possible by various techniques.

The velocity computation is a computer intensive process but it has to update the system fast enough to react the control thrusters accordingly. This calls for image matching algorithm which should work in real time. But image matching is challenging in case of varying lighting condition as well as Lander Dynamics which lunar lander could experience. This makes even the test and simulation also very difficult.

To compute velocity from images, image matching has to be performed between present frame to next frame. This can be done using spatial matching, feature based cross correlation etc. But the use of phase information of images i.e., correlation in phase domain which is feature independent can be superior option when compared with other methods. The peak in correlation function is likely to be detected more accurately using phase correlation method than the classical cross correlation [8-9]. Phase correlation gives a sharp peak at the registration point independent of lighting variation while cross correlation results in broad peaks with multiple side peaks. Lander dynamics as well as sample images can be stored for study and performance of the algorithm for post landing scenario. Here, in this paper a novel method with phase correlation implemented in FPGA including image storage unit is presented. The paper is organized as follows: In section II, overview of the sensor is given. The working of camera which is used to capture images is narrated in section III. Implementation of algorithm in FPGA is explained in section IV. The lab test results, aircraft test results and linear translator analyses of results are presented in section V. Finally, the inference drawn from the work done and future scope of work is discussed in section VI.

II. Overview

The System basically contains an Active Pixel Sensor Driven by FPGA which contains logic to drive APS, take the image data, and process it to compute pixel shift and transmit it to NGC (Navigation, guidance and control) using 1553B Interface. It also stores images in during landing in a SD flash and retrieved back post landing. Fig. 1 shows the block diagram of the sensor where RTAX2000S FPGA is interfaced

978-1-5386-3004-4/17/$31.00 ©2017 IEEE

with the APS detector, SD Flash, Oscillator, SRAM, NGC (Navigation guidance and control) and BDH (Baseband data handling). APS detector operates at 3 MHz; derived from a System clock of 24Mhz. FPGA generates the required drive signal as well as the address to drive the detector. The lunar terrain images are sensed by APS detector (image sensor). 8-bit pixel data is read by FPGA from detector. The acquired image data is having two different flows. In one flow, the velocity is computed from pixel shift between image sequences. Another flow, image frame is continuously stored.

For computing pixel shift between frames, two images need to be captured at a predefined interval. There are standard methods like feature extraction [2-3]. Feature matching is done so that pixel shift can be found. But usually feature extraction is very computation expensive process. This may be unsuitable for real time lander application. Besides feature based method [4] can derive maximum 17-20 pixel shift and it is more vulnerable to illumination variation and rotation. So a rather more robust method is used to find pixel shift through phase correlation in frequency domain [5].

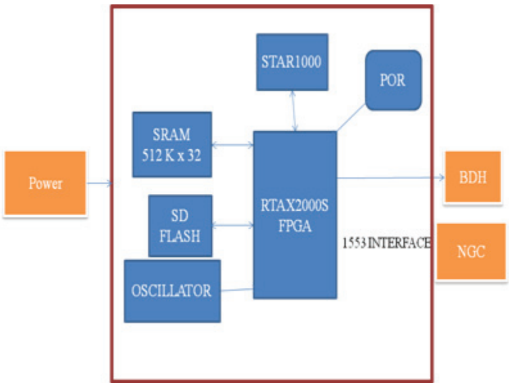

Figure 1 basic block diagram of lander horizontal velocity sensor

Pixel Shift in spatial domain causes phase change in frequency domain. This phase correlation method has less effect of illumination change as well as spacecraft rotation and can work up to 50% overlap between frames. It gives good result even for very dark and very bright images. The algorithm followed is explained using the flowchart given in fig.2.

III. DESIGN OF CAMERA ELECTRONICS

The lander camera is made out of Active Pixel Sensor detector which follows CMOS architecture. This requires low power to drive it as compared to CCD and involves direct FPGA driving. This makes the overall system light and less power consuming which is very useful for Space application as mass and volume is critical here. The detector STAR1000 is a Radiation HARD device which can work with harsh space environment in lunar environment in presence of Radiation. This detector works in a rolling shutter mode with maximum speed of 12MHz. With dating it is being operated with 3MHz rate and further windowing is possible to improve the update rate. Additionally, gain can be provided to improve the contrast performance from the detector. The above detector is already used in Navigation camera for Lunar Rover.

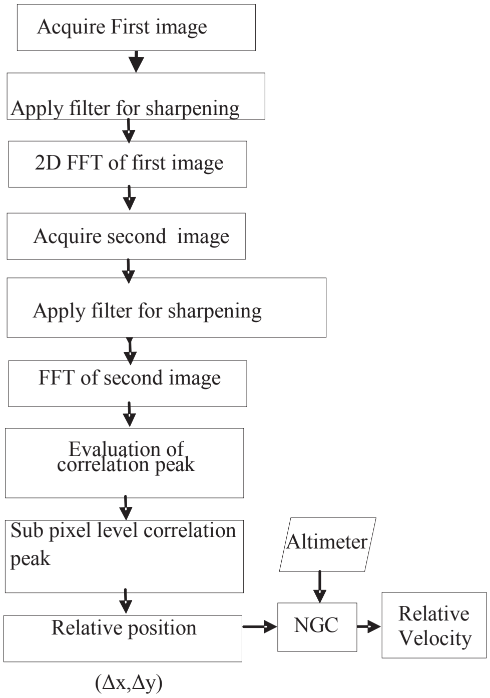

Figure. 2 Flowchart of the algorithm

Figure 3 LHVC electrical model

The ACTEL RTAX2000S FPGA drives the detector, process the images for shift and sends the data out to NGC. According to the algorithm, after capturing first image its 2D FFT is computed which are the result of 1D FFT row wise and then 1D FFT column wise.This is performed by an efficient high speed CoreFFT IP core. Then FFT results of both the images are multiplied, normalized and IFFT is computed. In the final result, IFFT correlation peak is found out. For better accuracy the peaks are derived with sub pixel accuracy [6-8].

The blocks are used for images processing are described below.

A. Image Preprocessing

Image captured from detector is required to do preprocessing before sending to FFT block for phase correlation. If Image involves blurring then phase correlation will fail. So Image sharpening filter has been incorporated in the front end of operation. Here vertical and horizontally image intensity differences are send to FFT block for computation.

B. 2D FFT computation

In the implementation of this method, phase correlation has to be computed in order to find pixel shifts between the frames. For both the images 2D FFT is computed .Then 2D FFT values are element wise conjugate multiplication is done and then they are normalized. 2D IFFT is performed on normalized values and bring back in spatial domain. This output contains a single peak for a constant shift in images. This peak position will detect pixel shift . Further sub pixel level peaks can be found using parabolic interpolation method.

FFT is implemented in FPGA using Actel soft IP core(figure 4). The general expression of Fourier Transform is the following

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}$$

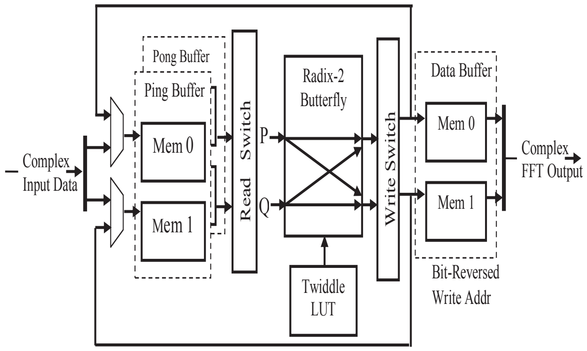

Figure 4 Actel Core FFT block diagram

If N is a power of 2 then the entire processing can be decomposed by addition and subtraction followed by multiplication with twiddle factor. This reduces the time complexity to N*log$_2$N. Input and output can be 8 bit real and imaginary data but processing can be done in 16 bit resolution to avoid finite word length effect.

FFT blocks will introduce scaling effect every block and therefore reducing the effective resolution. So to handle it, selective scaling can be incorporated as option. Intermediate results are stored in on-chip RAM which is made protected with EDAC (Error Detection and Correction) for this application. LUT based twiddle factor generation is used for FFT generation. IP core for FFT blocks are re used first for Row then for column to achieve high speed 2D FFT computation.FFT blocks have control over input and output Data rate. So initially it is synchronized with detector or memory based on the source of Data. With exact synchronism FFT of image is performed.

C. Normalization

After two FFT are multiplied, normalization is involved. CORDIC IP core is used to find the square root of a number using vectoring mode. CORDIC (Co-ordinate Rotation Digital computer) is an algorithm used to find vector rotation by a known angle using a LUT based method. This involves using only addition or subtraction in iterative way to find the magnitude and phase after rotation. During this iterative process, Cordic gain will be involved which will make scaled version of square root of a number An Actel FPGA optimized CORDIC engine for various purposes is CoreCORDIC. In CORDIC finding square root involves a vector rotation by 45 degree and the required operations are as shown in Figure 5 where K is CORDIC gain. Since it is a constant gain final IFFT results wont have effect due to this gain.

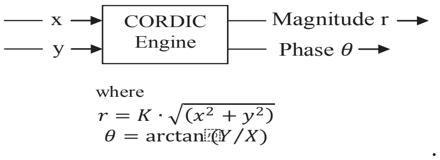

Figure. 5.CORDIC engine in vector mode

D. Division

For normalization purpose and during sub pixel computation division block will be required. The hardware implementation is mentioned below. Here, the divisor is shift-positioned and subtracted from the dividend. If a negative result is produced from subtraction of the divisor at any bit position relative to the dividend, the operation at that bit position is aborted and a 0 is placed in the respective location of the quotient. The divisor is restored to the result of the division operation, then the next highest bit of the dividend is shifted into the left bit position of the result. As each bit ofthe dividend is shifted from right to left, the quotient is built up from left to right.The division operation is complete, after N shifts, where N represents the number of bits in the dividend. This division is pipelined with CORDIC block to improve through put.

E. 2D IFFT computation

After Normalization, IFFT is performed on this normalized data using Core FFT block. Since it is 16 bit fixed point IFFT, to retain resolution, suitable multiplier has been used before doing IFFT. In this IFFT matrix, peak value location is found out and that location will give the pixel shift in x &y. For practical images , this correlation peak may not reside on integer pixel. So to derive pixel shift in sub pixel accuracy, parabolic interpolation in 2D domain is used.Parabolic interpolation is easy to implement in VHDL since it does not involve trigonometric or logarithm function.

F:Storage in SD Card :

The another flow is implemented to save the image data in a SD flash card in real time. SD flash memory is non volatile; therefore it retains data even after power off. SD mode consists of Command Response protocol in same bus. First, 48 bit Command is sent which includes Start BIT, Command Code, Parameter and CRC7 bits for error detection. A card ID is received as response from the Card. This Card ID can be used further to select the Card. Data storage can happen at 24 MHz rate.

Here for Data storage of one block (512 bytes), it may take some time before it accepts new data. But as Sensor data is available already efficient buffering is required to avoid losing any image data. Here basically FPGA internal FIFO is used and Read and Write maintains a fixed offset. So Buffer overrun never happens. Data are protected by CRC16.Here in 1GB SD flash , Datas are written continuously at a predefined rate (presently planned 2 frames in a second). Post Landing this image data can be read by command. In the command, starting address and number of frame are mentioned. So the required image data can be sent serially to ground frame by frame. Each frame will be time tagged so that its acquisition time reference can be obtained. These images can form a real landing video and can help understanding lander dynamics for future mission. These images could be very good test input for futuremission and the further optimization of velocity computation algorithm is possible.

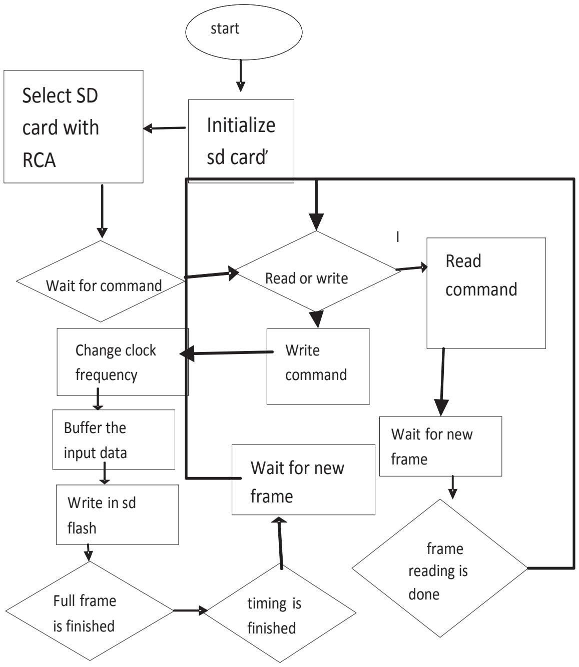

Figure 6 SD card reading and writing based on command

V. TEST SET UP

To test the algorithm in lab environment, LRO images are used. Videos generated from LRO images are run on a PC monitor and same video is used to derive the velocity.The video can be run at different frame rate which will simulate the velocity variation. Also direction also can be made reverse. On actual lunar landing, lander will move and here similar effect is generated as PC screen was kept at a known.distance and thus expected velocity is known and it has been matched with the velocity derived from the algorithm which computation has been done.Figure 6a and figure 6b are such two consecutive images on which computation has been done.After doing phase correlation\single strong peak is observed in correlation

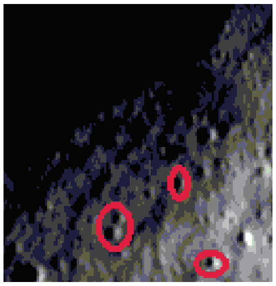

Figure 7a first frame in image sequence

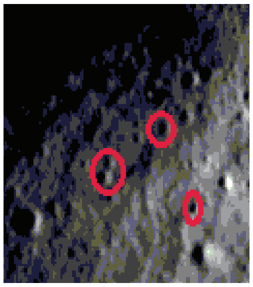

Figure 7b consecutive frames in image sequence

Figure7c correlation peak

matrix.This method can work for varying lighting condition and it generates very sharp peak which made it very robust.

.After doing phase correlation\single strong peak is observed in correlation matrix. This method can work for varying lighting condition and it generates very sharp peak which made it very robust. In this method a scale down simulation in lab can be performed. By knowing the frame rate, screen resolution and distance expected velocities are computed. Later on same test was repeated by a linear translator with fixed lunar images.

Using this test set up a comparison between MATLAB and VHDL post layout simulaton was done and they are matching closely .(RMS error of the order of 0.008).

A. *Aircraft test:*

Aircraft with GPS and IMU(Inertial measurement unit) was flown with Lander Camera with SD card on board with commercial optics. This will simulate the landing scenario at different altitude with known horizontal velocity. Thus using this test it is possible to simulate lunar lander actual scenario in much controlled and limited manner. However in this test due to blurred image velocity did not came on board. But with stored images, with additional filtering velocity was obtained and it is matched with GPS data. The aircraft was flown in different altitude (7.5km,4.2km,3km)etc and moved in a constant velocity. In the plot red lines are aircraft velocity from GPS data and blue lines are sensor output. They are synchronized with time within 0.5sec accuracy. It was matching within 2.5m/s. The following plots are there for 7.5km(figure 8a) and 4.2km(figure 8b) altitude. This testing provides closed resemblance with actual landing since there also Lander will move with some horizontal velocity before touchdown and slowly that velocity will be killed by thursters.With better sub pixel accuracy implemented in VHDL, better velocity accuracy is obtained. For real images, input image sharpening filter is put to overcome blurring problem which came because of usage of commercial optics for aircraft testing.

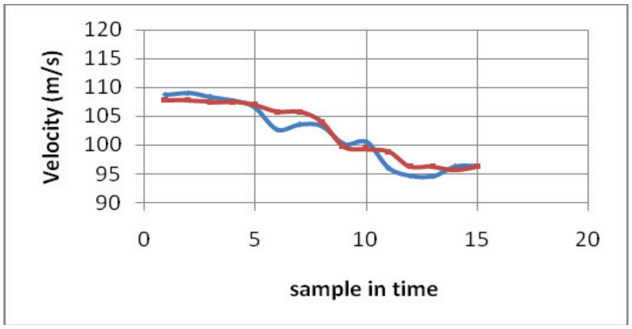

Figure 8a velocity plot at 7.5km altitude

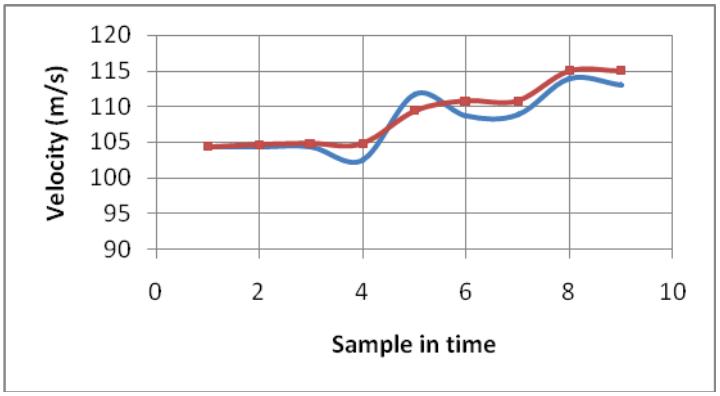

Figure 8b velocity plot at 4.2km altitude

B. *Linear translator test:*

In this set up , camera is mounted on a linear translator and printed lunar images pasted on wall. Velocities are known from translator settings and the same velocity will be obtained using the Lander Camera.The two plots are error plot when translator was moving with 20mm/sec(figure 9a) and 25mm/sec(figure 9b) velocity respectively. Here less than +/- 1mm/sec velocity accuracy is obtained. Using this set up polarity of velocity with respect to camera frame is also verified .Similar type of testing was done with outside real images also and results are matching.

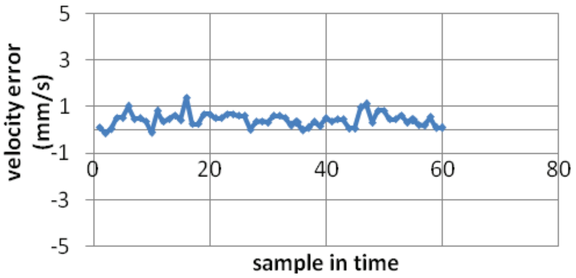

Figure 9a error plot with 25mm/sec velocity in linear translator

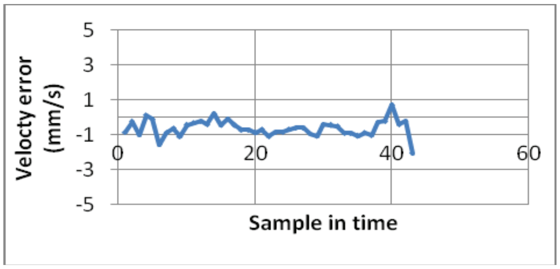

Figure 9b error plot with 20mm/sec velocity in linear translator

VI. CONCLUSION

The horizontal velocity estimation of a lunar lander with reference to terrain using phase correlation method with filtering is implemented in FPGA. The correctness was assured by verifying on different set of image pairs with varying intensity and contrast. The feature based method required distinct feature availability and overlap of same feature between two frames. It relies highly on thresholding, lighting condition and also consumes more time for computation. Feature based method requires SNR value of at least 50dB. Phase correlation technique can work with relatively less SNR and is invariant to low lighting conditions and small rotation. This was verified by testing also. Various noise types like uniform variations of illumination, average intensity offsets, and fixed gain errors due to calibration are overcome in phase correlation method. It can also handle small amount of blurring due to motion. Though a preprocessing filter will help to improve it. It is more effective and robust in intriguing conditions also..In aircraft images as Here, only translation is considered, the algorithm can be enhanced for better results by rotation and scaling correction in images.

ACKNOWLEDGMENT

The research work has been supported by Director, Laboratory for Electro-Optics Systems (LEOS)-ISRO, Dr. G Nagendra Rao. We also thank Associate Director, LEOS, A.S. Laxmi Prasad for his valuable encouragement and guidance.

REFERENCES

[1] G Ruicheng Yan, Zhiguo Cao, Jianhu Wang, Sheng Zhong, Dominik Klein, Armin Cremers, “Horizontal Velocity Estimation Via Downward Looking Descent Images for Lunar Landing”, IEEE transactions on Aerospace and Electronic Systems VOL. 50, NO. 2, APRIL 2014.

[2] Larry matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Wilson, Carlos Villalpando, Steve Goldberg, and Andrews Huertas, Jet Propulsion Laboratory, California Institute of Technology, USA, “Computer Vision on Mars”, International Journal of Computer Vision, 2007.

[3] Weill JIAO, Yaling FANG, Guojin HE. “An integrated feature based method for sub-pixel image matching”. *Center for Earth Observation and Digital Earth*, Chinese Academy of Sciences, CHINA.

[4] Harris, C., Stephens, M.., 1988. “A combined corner and edge detector”. In: fourth Alvey Vision Conference, Manchester, UK, pp.147-151.

[5] Zitova, B., Flusser, J., “Image registration methods:a survey”, Image and vision computing, 21, pp.977-1000, 2003

[6] Misganu Debella-Gilo, Andreas Kääb. “Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation". Institute of eco sciences, University of Oslo, Norway.

[7] Andrew Johnson, Reg Willson, Yang Cheng, Jay Goguen, Chris Leger, Miguel Sanmartin and Larry Matthies , Jet Propulsion Laboratory, California Institute of Technology, CA, USA “Design Through Operation of an Image-Based Velocity Estimation System for Mars Landing”, International Journal of Computer Vision,2007.

[8] Hassan Foroosh (Shekarforoush), Josiane B.Zerubia, ”Extension of Phase Correlation to Subpixel Registration”, IEEE transactions on Image Processing,Vol.11, No.3,March 2002.

[9] Arindam Mal et. Al “Vision based navigation for safe landing in interplanetary missions”, *LCPM-10*, June 2013, USA.